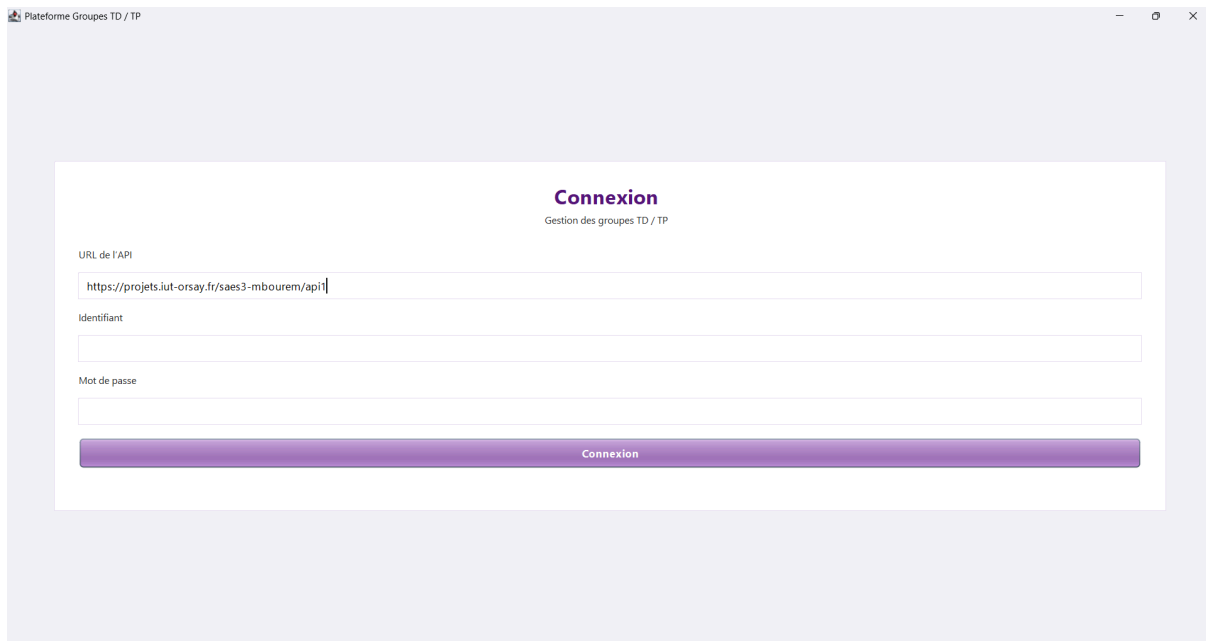


# Partie Qualité du Développement Objet (R304)

## 1. Page de connexion



Plateforme Groupes TD / TP

**Connexion**  
Gestion des groupes TD / TP

URL de l'API

Identifiant

Mot de passe

**Connexion**

Pour se connecter à l'application il faut avoir un login et un mot de passe. Quand je clique sur connexion ça envoie une requête à l'API : "POST /auth/login" avec un body en json ou il y a écrit mon login et mon mot de passe. Exemple :

```
{  
  "login": "responsable",  
  "password": "test123"  
}
```

et l'API va vérifier qu'il existe bien un utilisateur avec ce login et ce mot de passe dans la base de données

Une fois connecté, une première requête s'exécute automatiquement pour récupérer et afficher tous les étudiants du S2. Pour voir tous les étudiants du S3, il faut le sélectionner en haut à gauche. Cela change car la requête vers l'API n'est plus la même. On prend l'exemple des étudiants en BUT 1 en semestre 1

Exemple : GET /formations/1/semestres/1/etudiants

Ce qui va renvoyer un fichier JSON du type, pour 2 étudiants pas exemple :

```
[
{
  "idEtu": 28,
  "nomEtu": "Amara",
  "prenomEtu": "Adama",
  "genreEtu": 1,
  "typeBac": "Techno",
  "indiceCovoiturage": 8,
  "estAnglophone": 1,
  "estRedoublant": 1,
  "estApprenti": 0,
  "nomGroupe": null,
  "emailUniEtu": "adama.amara28@universite.fr"
},
{
  "idEtu": 29,
  "nomEtu": "Bernard",
  "prenomEtu": "Sofia",
  "genreEtu": 1,
  "typeBac": "General",
  "indiceCovoiturage": 0,
  "estAnglophone": 0,
  "estRedoublant": 0,
  "estApprenti": 0,
  "nomGroupe": "G3",
  "emailUniEtu": "sofia.bernard29@universite.fr"
}
]
```

SAÉ - Création de groupes TD/TP
Connecté : responsable (RESP\_FORMATION)
Formation : 1 - BUT Informatique
Semestre : S2
Recharger
Déconnexion

Manuel
Automatique

**Gestion manuelle des groupes**  
Affectation, consultation et gestion des groupes

Rafraîchir
Nouveau groupe : 
Créer groupe

**Étudiants**  
Sélectionne un étudiant

ID	Nom	Prénom	Bac	Genre	Redoublant	Covoit	Groupe
1044	Amara	Adama	Techno	1	Oui	8	G2
1107	Amara	Omar	STD2A	1	Oui	28	G7
1123	Amara	Omar	STI2D	1	Non	33	G5
1131	Amara	Yanis	Techno	1	Non	35	G5
1115	Barbier	Luan	STI2D	1	Non	30	G5
1026	Barbier	Omar	Général	1	Non	4	G1
1060	Benoit	Celia	STD2A	0	Non	14	G3
1069	Benoit	Jade	Techno	0	Non	16	G3
1138	Benoit	Julien	Général	1	Oui	38	G8
1033	Benoit	Rafael	STD2A	1	Non	0	G9
1043	Benoit	Tom	Général	1	Non	8	G2
1162	Bernard	Elliott	STD2A	1	Non	46	G10
1051	Bernard	Hugo	STI2D	1	Non	10	G3
1156	Bernard	Ibrahim	STMG	1	Non	44	G10
1133	Bernard	Karim	STMG	1	Non	36	G8
1155	Bernard	Ruben	Général	1	Oui	44	G10
1106	Bernard	Sofia	STD2A	0	Non	28	G7
1121	Bonnet	Armand	STD2A	1	Non	32	G7
1039	Bonnet	Lior	Général	1	Non	7	G2
1065	Chevalier	Leo	STD2A	1	Non	15	G3

**Détails étudiant**  
Informations détaillées

Sélectionne un étudiant pour voir ses informations.

**Groupes**  
Gestion des groupes

Groupe	Effectif
G1	15
G10	13
G2	15
G3	15
G4	15
G5	14
G6	14

Ajouter
Retirer
Voir membres
Supprimer

On a maintenant plusieurs possibilités qui s'offrent à nous.

## Groupes

Gestion des groupes

Groupe	Effectif
G1	16
G10	13
G2	16
G3	15
G4	15
G5	14
G6	14
G7	14

Ajouter

Retirer

Voir membres

Supprimer

On va commencer par supprimer le groupe G10 car on l'a créé sans le faire exprès. Quand on clique sur supprimer, la requête :

"DELETE /formations/1/semestres/1/groupes/G10" est envoyé à l'API. l'API essaie de supprimer dans la base avec une requête sql et si il n'y a pas de contraintes le groupe est supprimé, sinon un message d'erreur est envoyé.

On peut aussi retirer ou ajouter un étudiant à un groupe. Il suffit de choisir un étudiant et un groupe puis cela va envoyer une requête à l'API :

"Put /etudiants/1/groupe". l'API va récupérer le body dans l'URL pour créer un nouvel étudiant dans la base avec une requête sql.

Et pour retirer il suffit de mettre null au niveau du groupe. Pour ce faire il faut exécuter cette requête : "Put /etudiants/1/groupe" et l'API va instancier nomGroupe à null.

SAÉ - Création de groupes TD/TP

Connecté : responsable (RESP\_FORMATION)

Formation : 1 - BUT Informatique Semestre : S3 [Recharger] [Déconnexion]

Manuel Automatique

### Gestion manuelle des groupes

Affectation, consultation et gestion des groupes

[Rafraîchir] Nouveau groupe : [ ] [Créer groupe]

#### Étudiants

Sélectionne un étudiant

ID	Nom	Prénom	Bac	Genre	Redoublant	Covoit	Groupe
428	Amara	Adama	Techno	1	Oui	8	G16
491	Amara	Omar	STD2A	1	Oui	28	G16
507	Amara	Omar	STI2D	1	Non	33	G16
515	Amara	Yanis	Techno	1	Non	35	G16
499	Barbier	Luan	STI2D	1	Non	30	G16
410	Barbier	Omar	Général	1	Non	4	G2
444	Benoit	Celia	STD2A	0	Non	14	G16
453	Benoit	Jade	Techno	0	Non	16	G16
522	Benoit	Julien	Général	1	Oui	38	G2
417	Benoit	Rafael	STD2A	1	Non	0	G2
427	Benoit	Tom	Général	1	Non	8	G2
546	Bernard	Elliott	STD2A	1	Non	46	G2
435	Bernard	Hugo	STI2D	1	Non	10	G3
540	Bernard	Ibrahim	STMG	1	Non	44	G3
517	Bernard	Karim	STMG	1	Non	36	G3
539	Bernard	Ruben	Général	1	Oui	44	G3
490	Bernard	Sofia	STD2A	0	Non	28	G16
505	Bonnet	Armand	STD2A	1	Non	32	G3
423	Bonnet	Lior	Général	1	Non	7	-
449	Chevalier	Leo	STD2A	1	Non	15	-

#### Détails étudiant

Informations détaillées

ID : 453  
 Nom : Benoit  
 Prénom : Jade  
 Email : jade.benoit53@universite-paris-saclay.fr3  
 Bac : Techno  
 Genre : M  
 Redoublant : Non  
 Anglophone : Non  
 Apprenti : Non

#### Groupes

Gestion des groupes

Groupe	Effectif
G14	14
G16	14
G17	13
G18	14
G19	14
G2	13
G3	5
G9	5

[Ajouter] [Retirer] [Voir membres] [Supprimer]

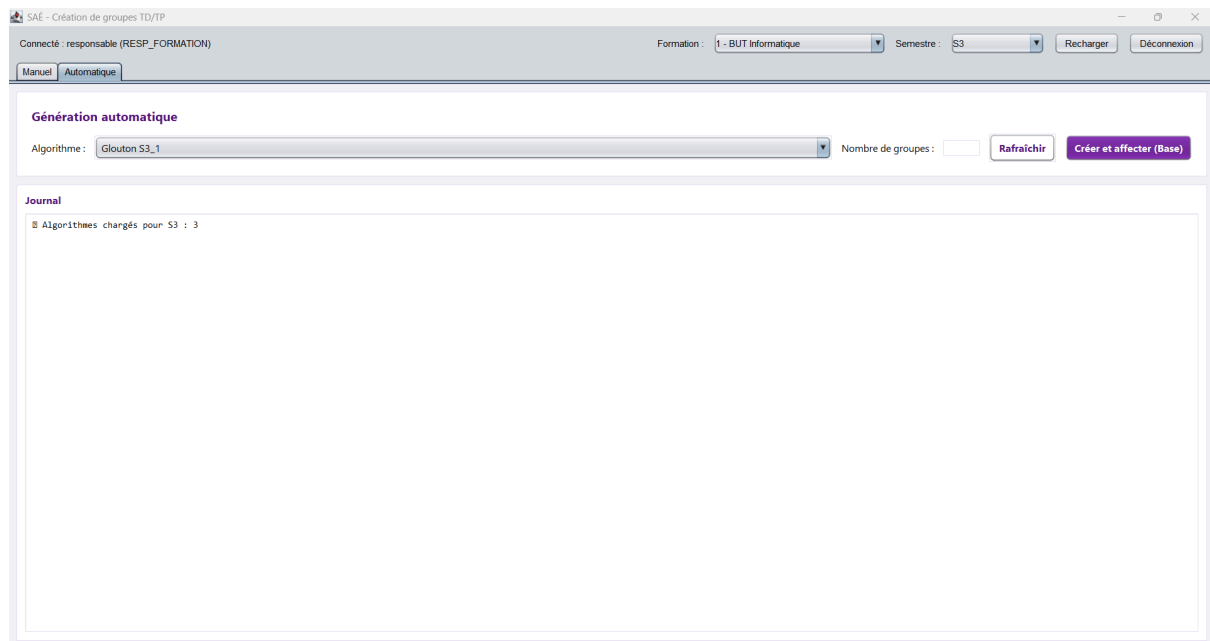
On peut aussi créer des groupes manuellement avec une requête du type :

“Post /groupes” avec comme body :

```
{
  "idFormation": 1,
  "numSemestre": 1,
  "nomGroupe": "G12"
}
```

l'API va donc rajouter ce groupe et renvoyé un JSON pour confirmé ou non la création du groupe

```
{
  "ok": true,
  "groupe": {
    "idFormation": 1,
    "numSemestre": 1,
    "nomGroupe": "G12"
  }
}
```

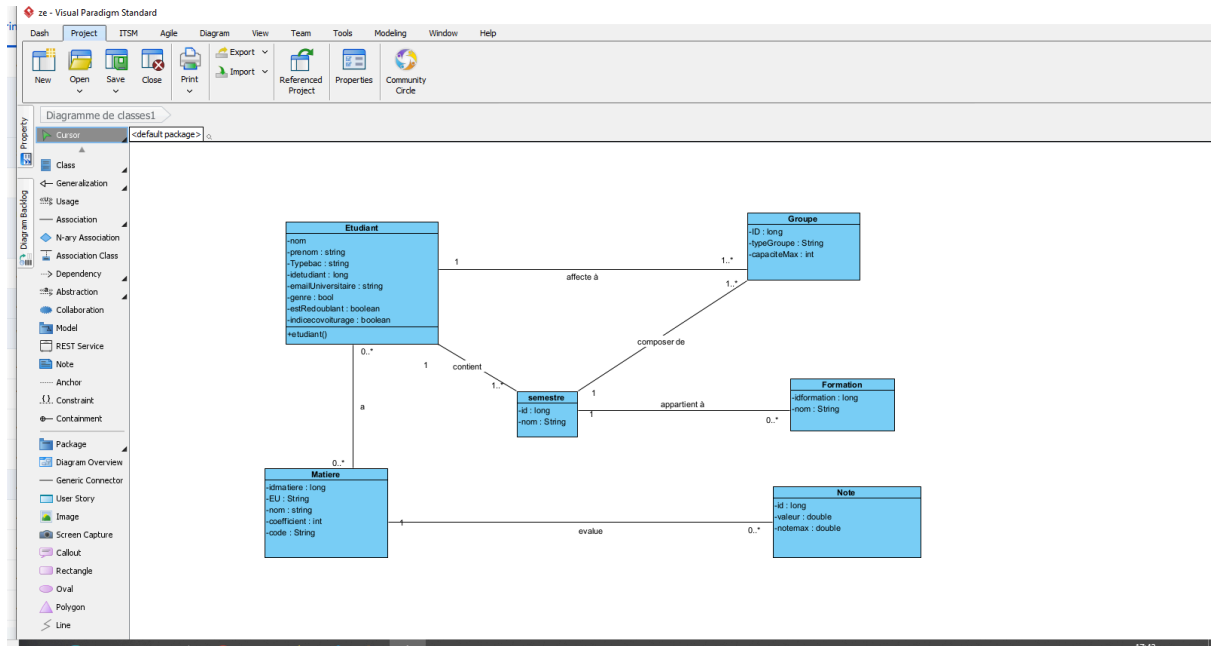


La deuxième permet d'attribuer des groupes aux étudiants qui n'en n'ont pas avec différents algorithmes à disposition.

quand on va exécuter les différents algorithmes cela va directement attribuer aux étudiants un groupe grâce à des requêtes.

Quand j'exécute un algorithme (glouton ou brute force) dans l'application Java, l'algorithme ne tourne pas sur le serveur : il tourne en local dans Java à partir de la liste d'étudiants récupérée via l'API. Concrètement, l'application envoie d'abord des requêtes GET pour charger les données (par exemple la liste des étudiants d'un semestre et les groupes existants). Ensuite, une fois la solution calculée, l'application écrit le résultat dans la base en appelant l'API : elle envoie une requête POST pour créer chaque groupe, puis des requêtes PATCH pour affecter chaque étudiant à un groupe (ou le retirer en mettant `nomGroupe` à `null`). C'est donc bien l'API qui modifie la base de données, mais uniquement après que l'algo Java a produit la répartition.

## Diagramme de classe



## Diagramme de déploiement

